

Programación—Certamen Recuperativo - 21 de Agosto de 2018

Nombre

Rol

Paralelo

2. [40 %] **NOTA:** Puede hacer uso de las funciones definidas en la pregunta 1.

Pythonmed es un centro médico donde se puede solicitar una hora de consulta con distintos especialistas. Considere la lista de tuplas `medicos`, donde cada tupla de la lista contiene el rut, nombre y especialidad de un médico.

```
medicos = [('18861898-8', 'Daniel Sard', 'Medicina General'),
           ('17302713-1', 'Sebastian Moreno', 'Cardiologia'),
           ('15361593-2', 'Armando Rossat', 'Pediatria'),
           ('13594298-4', 'Luis Montano', 'Dermatologia'), ...]
```

Por otra parte, los horarios de atención se almacenan en el diccionario `agenda`. Cada llave de esta estructura corresponde al rut de un médico y el valor es un conjunto de tuplas con información de los horarios de atención. Considere la información de horarios de atención como tuplas con un string de la fecha y hora en el formato `AAAAMMDDHHMM` y un string que indica disponibilidad (O: libre, X: ocupado). Los puntos suspensivos en los ejemplos de las estructuras presentadas, señalan que pueden existir muchos más registros, por lo que **asuma que hay muchos más datos que los presentados**.

```
agenda = {'13594298-4': {('201807231400', 'X'), ('201808111500', 'X'),
                        ('201807231430', 'O'), ... }, '17302713-1': {('201807211400', 'X'),
                        ('201807261100', 'O'), ('201808161430', 'O')}, ... }
```

- a) Escriba la función `agenda_por_dia(agenda, rut, fecha)` que reciba la estructura `agenda`, un string con el rut de un médico y una fecha en el formato `AAAAMMDD`. La función debe retornar **una lista de tuplas** con los horarios de atención para el rut y fecha ingresados como parámetro. Esta lista deberá estar **ordenada por hora en forma ascendente**. Asuma que el rut del médico ingresado, siempre existirá y estará correcto.

```
>>> agenda_por_dia(agenda, "13594298-4", "20180723")
[... , ('201807231400', 'X'), ('201807231430', 'O'),
 ('201807231500', 'X'), ...]
```

- b) Escriba la función `solicitar_hora(agenda, rut, horario)` que reciba como parámetros la estructura `agenda`, un string con el rut del médico y un string con el horario de atención en el formato `AAAAMMDDHHMM`. La función **debe verificar si hay disponibilidad de atención del médico** para el horario ingresado y retornar la agenda actualizada con el horario de atención ocupado. Si no existe el horario ingresado o está ocupado debe retornar `False`. Asuma que el rut del médico ingresado, siempre existirá y estará correcto.

```
>>> solicitar_hora(agenda, "13594298-4", "201807231400")
False
>>> solicitar_hora(agenda, "13594298-4", "201807231430")
{'13594298-4': set([('201807231400', 'X'), ('201808111500', 'X'),
                  ('201807231430', 'X'), ... ]), ...}
```

